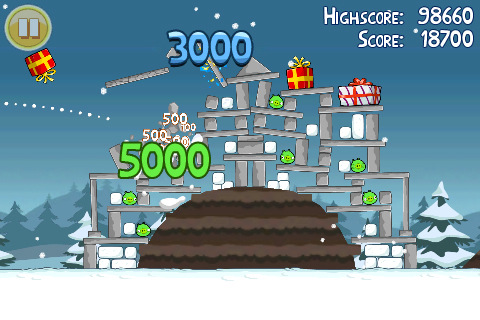
The game I’m would be making is called ….. The game is make by half brick, but instead of using?????







# Game idea

The game idea is to incorporate the successfully elements of are Fish out of Water, Angry Bird and Fruit Ninja. The objective of this game is to get the highest score and also it is also an endless game. To able to get the highest score the player would need to damage as much object as we can. In the beginning of the level the player will need to drag a fish out of water, after that the player will need to shoot the fish to the direction that which where the object are the fish inside contains a bomb so which allow the fish splice into two piece and it will lead the bomb to expose. The exposition will hit the objects in a range and depends what object is has it will get a score. This project will also aim to have multi levels.

# Game Value ( I will deal with this one)

The reason

Why not use other

They are popular games which means the download rate is high

2. Wanted to go to half-brick after finish study

Why if I put it together why does it work

If it just combines elements how does it work?

Why still want to use it

The market people market:

Why is it best these people?

Use as adv.

I will mixed up the concept of three games which are Fish out of Water, Angry Bird and Fruit Ninja, There are three main reason behind it.

1. They are popular games which means the download rate is high

2. Wanted to go to half-brick after finish study

3.

# Major challenge

## Challenge and Solution (TOUCH ScREEN)

One of the biggest challenge I may face will be because of my lack of experience with touch screen technology

The able solve this solution for this challenge.

One of the method is list of video which come from YouTube:

<https://www.youtube.com/results?search_query=unity+mobile+control+tutorial>

<https://unity3d.com/learn/tutorials/projects/2d-roguelike/mobile>

<https://unity3d.com/learn/tutorials/modules/beginner/live-training-archive/mobile-development>

<http://imakeinternet.com/unity-2d-tutorial-screen-player-touch-controller/>

## Challenge and Solution (WATER)

The 2 one will be ocean the how the animals interacts when out of the water and in the water. The challenge of this part is because when the fish goes into different layer they will have different behaviors and I have no idea of how the fishes go to work in the ocean. The following link

<http://gamedevelopment.tutsplus.com/tutorials/make-a-splash-with-dynamic-2d-water-effects--gamedev-236>

<http://gamedevelopment.tutsplus.com/tutorials/creating-dynamic-2d-water-effects-in-unity--gamedev-14143>

## Challenge and Solution (Random ALGORITHM)

Also the 3 part is setting up all the level with all the objects, since I don’t know do I have enough time for the first level, but aiming to create multiple levels .So creating multiple level we would need to use some Map Generation Algorithm, following by the list below:

* [Artificial Life](http://pcg.wikidot.com/pcg-algorithm:artificial-life)
* [Caves](http://pcg.wikidot.com/pcg-algorithm:caves)
* [Cellular Automata](http://pcg.wikidot.com/pcg-algorithm:cellular-automata)
* [City Generation](http://pcg.wikidot.com/pcg-algorithm:city-generation)
* [Cubic Sphere](http://pcg.wikidot.com/pcg-algorithm:cubic-sphere)
* [Diamond-Square Algorithm](http://pcg.wikidot.com/pcg-algorithm:diamond-square-algorithm)
* [Drunkard Walk](http://pcg.wikidot.com/pcg-algorithm:drunkard-walk)
* [Dungeon Generation](http://pcg.wikidot.com/pcg-algorithm:dungeon-generation)
* [Dynamic Weather](http://pcg.wikidot.com/pcg-algorithm:dynamic-weather)
* [Fire Propagation](http://pcg.wikidot.com/pcg-algorithm:fire-propagation)
* [Fluid Dynamics](http://pcg.wikidot.com/pcg-algorithm:fluid-dynamics)
* [Forests](http://pcg.wikidot.com/pcg-algorithm:forests)
* [Fractal](http://pcg.wikidot.com/pcg-algorithm:fractal)
* [Fractal River Basins](http://pcg.wikidot.com/pcg-algorithm:fractal-river-basins)
* [Fractional Brownian motion](http://pcg.wikidot.com/pcg-algorithm:fractional-brownian-motion)
* [Genetic Algorithm](http://pcg.wikidot.com/pcg-algorithm:genetic-algorithm)
* [Height Maps](http://pcg.wikidot.com/pcg-algorithm:heightmap)
* [Iterated Function System](http://pcg.wikidot.com/pcg-algorithm:iterated-function-system)
* [L-System](http://pcg.wikidot.com/pcg-algorithm:l-system)
* [Mazes](http://pcg.wikidot.com/pcg-algorithm:maze)
* [MegaTexture](http://pcg.wikidot.com/pcg-algorithm:megatexture)
* [Midpoint Displacement Algorithm](http://pcg.wikidot.com/pcg-algorithm:midpoint-displacement-algorithm)
* [Perlin Noise](http://pcg.wikidot.com/pcg-algorithm:perlin-noise)
* [Plant Generation](http://pcg.wikidot.com/pcg-algorithm:plant-generation)
* [Procedural Sky](http://pcg.wikidot.com/pcg-algorithm:procedural-sky)
* [Procedural Spooling](http://pcg.wikidot.com/pcg-algorithm:procedural-spooling)
* [Procedural Texture](http://pcg.wikidot.com/pcg-algorithm:procedural-texture)
* [Rapidly-exploring random tree](http://pcg.wikidot.com/pcg-algorithm:rapidly-exploring-random-tree)
* [Reaction-Diffusion System](http://pcg.wikidot.com/pcg-algorithm:reaction-diffusion-system)
* [Simplex Noise](http://pcg.wikidot.com/pcg-algorithm:simplex-noise)
* [Texture Synthesis](http://pcg.wikidot.com/pcg-algorithm:texture-synthesis)
* [Universe Generation](http://pcg.wikidot.com/pcg-algorithm:universe-generation)
* [Voronoi Diagram](http://pcg.wikidot.com/pcg-algorithm:voronoi-diagram)
* [Whittaker Diagram](http://pcg.wikidot.com/pcg-algorithm:whittaker-diagram)
* [Worley Noise](http://pcg.wikidot.com/pcg-algorithm:worley-noise)

This link comes from <http://pcg.wikidot.com/pcg-algorithm:map-generation>